J. Soc. Indust. Appl. Math.
Vol. 5, No. 2, June, 1957
Printed in U.S.A.

GENERATION AND USE OF ORTHOGONAL POLYNOMIALS FOR DATA-FITTING WITH A DIGITAL COMPUTER*

George E. Forsythe†

1. Introduction. Let $x_1, \cdots, x_\mu, \cdots, x_m$ be given values of an independent real variable x. Suppose that corresponding to each value x_μ a real number $f_\mu(\mu = 1, \cdots, m)$ is given. Here f_μ may be the observed value or computed value at x_μ of some function f of the independent variable x.

Suppose that it is desired to fit the data values $f_1, \cdots, f_m$ by a polynomial $y_k(x)$ of given degree k:

$$y_k(x) = t_0^{(k)} + t_1^{(k)}x + \cdots + t_k^{(k)}x^k. \tag{1}$$

By $y_k(x)$ *fitting the data* we mean roughly that

$$| y_k(x_\mu) - f_\mu | \quad \text{is small for each} \quad \mu = 1, \cdots, m. \tag{2}$$

There are many ways to make precise the vague conditions (2).

If $k + 1 \geqq m$, there are enough parameters $t_i^{(k)}$ so that $y_k(x_\mu) - f_\mu$ can be made 0 for each μ, and there is no problem in interpreting (2). The algorithm is then one of *polynomial interpolation*, in which the $t_i^{(k)}$ are selected to make $y_k(x)$ pass through each point (x_μ, f_μ). But when, as we shall henceforth suppose,

$$k + 1 < m, \tag{3}$$

the $y_k(x_\mu) - f_\mu$ cannot ordinarily all be made 0 simultaneously. When they cannot, the m conditions (2) compete with one another, and the numerical analyst must somehow take account of this in order to formulate a *problem of data-fitting.*

The m numbers $e_\mu = y_k(x_\mu) - f_\mu$ are the m components of an *error vector* e. Since the x_μ and f_μ are regarded as fixed, the vector e depends only on the parameters $t_0^{(k)}, \cdots, t_k^{(k)}$. Each common method for dealing with the competing requirements (2) corresponds to the selection of a *norm* $\| e \|$ for the vector e. The two norms most frequently considered are the *minimax norm*

$$\| e \|_\infty = \max_{\mu=1,\cdots,m} | e_\mu |$$

* Received by the editors March 2, 1957.

† The author's work on this paper at the University of California, Los Angeles, was sponsored by the Office of Naval Research. Reproduction in whole or in part is permitted for any purpose of the United States Government.

The author's work at The Ramo-Wooldridge Corporation was sponsored by the U. S. Air Force.

and the *euclidean norm*

$$\| e \|_2 = \{e_1^2 + \cdots + e_m^2\}^{\frac{1}{2}}. \tag{4}$$

For any choice of norm $\| e \|$, the mathematical problem of data-fitting is that of finding values of the parameters $t_0^{(k)}, \cdots, t_k^{(k)}$ so that

$$\| e \| = \text{minimum}. \tag{5}$$

The last sentence is the precise formulation of our problem.

The solution of the problem for the norm $\| e \|_\infty$ is very satisfactory in the following sense. It means that we have found a polynomial $y(x)$ and an $\epsilon > 0$ such that

$$| y(x_\mu) - f_\mu | \leqq \epsilon \qquad (\text{for all } \mu = 1, \cdots, m) \tag{6}$$

and that no $y_1(x)$ and $\epsilon_1 < \epsilon$ can be found such that (6) is satisfied for ϵ_1. Thus the polynomial $y(x)$ deviates from f_μ by more than ϵ at none of the arguments $x_1, \cdots, x_m$, while no $\epsilon' < \epsilon$ will have the same property. If the f_μ are known to be exact, such a fit is very desirable.

However, the numerical determination of the $y(x)$ corresponding to the norm $\| e \|_\infty$ is considerably more difficult than with the norm $\| e \|_2$, largely because the norm $\| e \|_\infty$ is not even a differentiable function of the components e_μ, nor of the t's, in the vicinity of the solution (see [1]). Hence the norm $\| e \|_2$ is more frequently chosen. Moreover, if it is assumed that the values f_μ are subject to independent normally distributed errors about their trend (due, for example, to observational errors) the use of the norm $\| e \|_2$ is demanded by regression theory (see [2]).

In the present note we shall develop the theory and practice of data-fitting according to the norm $\| e \|_2$. Although little of this material is new, the author has frequently been asked to write a self-contained presentation. The use of orthogonal polynomials in curve fitting is standard [10], but their numerical generation by the three-term recurrence (Section 6) seems to have been recommended only recently (see Lanczos [3], Householder [4], and Stiefel [5]). The relation of the normal equations to the Hilbert matrix was pointed out by Hilbert [12].

2. Normal equations. As was stated above, we seek $t_0^{(k)}, \cdots, t_k^{(k)}$ so that

$$\Phi(t_0^{(k)}, \cdots, t_k^{(k)}) = \sum_{\mu=1}^{m} \{f_\mu - y_k(x_\mu)\}^2 = \text{minimum}. \tag{7}$$

The reader who understands matrix notation should go at once to Section 9 for the complete derivation of (11). In this present section, the normal equations (11) will be derived by the use of calculus, but there will be a gap in the proof.

Substituting for $y_k(x_\mu)$ from (1), one finds that

$$\Phi = \sum_{\mu=1}^{m} \left\{ f_\mu - \sum_{h=0}^{k} t_h^{(k)} x_\mu{}^h \right\}^2 . \tag{8}$$

From (8) we see that Φ is a quadratic function of $t_0^{(k)}, \cdots, t_k^{(k)}$, and is therefore differentiable everywhere. Hence, if Φ has a minimum for $t_0^{(k)}, \cdots, t_k^{(k)}$, we will have

$$\frac{\partial \Phi}{\partial t_i^{(k)}} = 0 \qquad \text{(for all } i = 0, 1, \cdots, k).$$

But then

$$\frac{\partial \Phi}{\partial t_i^{(k)}} = 2 \sum_{\mu=1}^{m} \left\{ f_\mu - \sum_{h=0}^{k} t_h^{(k)} x_\mu{}^h \right\} (-x_\mu{}^i) = 0$$

$$\text{(for all } i = 0, \cdots, k).$$

Therefore, cancelling -2 and interchanging $\sum_\mu$ and $\sum_h$, we observe that at any minimum

$$\sum_{h=0}^{k} t_h^{(k)} \left\{ \sum_{\mu=1}^{m} x_\mu{}^h x_\mu{}^i \right\} = \sum_{\mu=1}^{m} f_\mu x_\mu{}^i \quad \text{(for all } i = 0, \cdots, k). \tag{9}$$

Let us introduce the abbreviations

$$g_{hi} = \sum_{\mu=1}^{m} x_\mu{}^h x_\mu{}^i = \sum_{\mu=1}^{m} x_\mu^{h+i}; \ \gamma_i = \sum_{\mu=1}^{m} f_\mu x_\mu{}^i . \tag{10}$$

Then we see from (9), (10) that at any minimum the $t_0{}^{(k)}, \cdots, t_k{}^{(k)}$ must satisfy the system of equations

$$\begin{array}{c} g_{00} t_0^{(k)} + \cdots + g_{0k} t_k^{(k)} = \gamma_0 \\ \cdots\cdots\cdots\cdots\cdots \\ g_{k0} t_0^{(k)} + \cdots + g_{kk} t_k^{(k)} = \gamma_k . \end{array} \tag{11}$$

The equations (11) are called the *normal equations* of the least-squares data-fitting problem which we are solving.

If the determinant $| g_{hi} |$ of the system (11) were known not to vanish, one would know there is a unique set of solutions $t_0^{(k)}, \cdots, t_k^{(k)}$. Since $\Phi = \| e \|_2^2 \geqq 0$, it would then seem plausible that such a unique solution of (11) would actually minimize Φ. The verification of the existence of a unique minimizing set $t_0^{(k)}, \cdots, t_k^{(k)}$ is easily carried out with matrix calculations and without differentiation, as is shown in Section 9.

With this qualification, we have shown that the approximation prcblem (7) has a unique solution which is obtained from (11).

3. Solving the normal equations. The equations (11) are easy to determine for a numerically given set of couples $(x_\mu, f_\mu)(\mu = 1, \cdots, m)$. Hence the numerical solution of (11) seems to be a straightforward way of solving the data-fitting problem (7). Experience shows that this procedure works very well for $k = 1, 2, \cdots$, up to perhaps 5 or 6. One gets the minimizing $t_0^{(k)}, \cdots, t_k^{(k)}$ relatively easily, and they can be used to compute $y_k(x_\mu)$.

When $k \geqq 7$ or 8, however, one begins to hear strange grumblings of discontent in the computing laboratory. The gist of the unhappiness is that each method selected to solve the system (11) fails somehow for the larger values of k. Let us therefore attempt to get a crude model of the system (11).

Suppose that m is large, and that the x_μ are distributed approximately uniformly on the interval (0, 1). Then we may expect that

$$\sum_{\mu=1}^{m} x_\mu^{\ h} x_\mu^{\ i} \approx m \int_0^1 x^h x^i \, dx = m \int_0^1 x^{h+i} \, dx = \frac{m}{h+i+1}.$$

Thus the coefficient matrix $G = (g_{hi})$ of the system (11) is something like m times the matrix $[(h + i + 1)^{-1}]$ $(h, i = 0, \cdots, k)$.

But the latter matrix is the well known and notorious principal minor of order $k + 1$ of the infinite *Hilbert matrix*

$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots \\ \vdots & \vdots & \vdots & \end{bmatrix}.$$

It has been observed frequently that systems of linear equations involving minors of H are very difficult to solve. For $k = 9$, for example, the order of the principal minor H_{10} is 10, and the inverse H_{10}^{-1} has elements of magnitude $3 \cdot 10^{12}$ (see [6]). Thus a slight error of 10^{-10} in one γ_i will lead to errors of approximately 300 in the $t_i^{(k)}$ corresponding to the solution of (11). All experience shows that it is very difficult to solve the system (11) with such a matrix.

With this model of the system (11), the grumblings in the computing laboratory become understandable.

4. Interpreting the solution by regression theory. Suppose, in spite of the difficulties developed in Section 3, that one could manage to solve the normal equation (11) for various values of k. How could one tell which was the correct value of k to use?

The usual point of view [2] is to assume that the $\{f_\mu\}$ are independently normally distributed about some polynomial trend $y_{h+1}(x) = \sum_{i=0}^{h+1} r_i x^i$ with a variance σ^2 independent of μ. Then one makes the *null hypothesis* that

$r_{h+1} = 0$, no matter what values $r_0, \cdots, r_h$ and σ^2 may have. The statistical test function for testing the hypothesis that $r_{h+1} = 0$ is a simple function of the σ_h^2 and σ_{h+1}^2 defined below.

Let $y_k(x)$ denote the polynomial of degree k which best fits the data f_μ in the sense of minimizing Φ in (7). Let

$$\delta_k^2 = \sum_{\mu=1}^{m} \{f_\mu - y_k(x_\mu)\}^2, \tag{12}$$

and let

$$\sigma_k^2 = \delta_k^2(m - k - 1)^{-1}. \tag{13}$$

Under the null hypothesis it follows [2] that the *expected value* of the statistic σ_k^2 is independent of k, for $k = h + 1, h + 2, \cdots, m - 2$.

In practice one therefore wants to compute σ_k^2 for $k = 1, 2, \cdots$. As long as σ_k^2 decreases significantly as k increases, one may presume that $y_k(x)$ is a valid fit to the data. If, after a certain value $k = k_1$ there is no further significant decrease in σ_k^2—i.e., if $\sigma_k^2 \approx \sigma_{k_1}^2$ for all $k \geqq k_1$—one may presume that the data f_μ are realistically represented by the polynomial $y_{k_1}(x)$. (Developing suitable tests for the σ_k^2 will require some experience.)

Note that for $k = m - 1$ one can pass $y_k(x)$ through any m ordinates $f_1, \cdots, f_m$, so that $\delta_{m-1}^2 = 0$. But σ_{m-1}^2 takes the meaningless form 0/0, showing how the decrease in the denominator of σ_k^2 compensates for the unavoidable decrease in δ_k^2.

To compute σ_k^2 one will first compute $y_k(x_\mu) = t_0^{(k)} + t_1^{(k)}x_\mu + \cdots + t_k^{(k)}x_\mu^k$ for each $\mu = 1, \cdots, m$. Here each $t_i^{(k)}$ depends both on i and on k. In computing δ_{k+1}^2, no practical use can be taken of the information already gained in computing δ_k^2, because there is no applicable relation between the $t_i^{(k)}$ and the $t_i^{(k+1)}$. Consequently, the computation of $\sigma_1^2, \sigma_2^2, \cdots, \sigma_r^2$ will require r independent polynomial fits, followed by independent calculations of σ_k^2 from (12) and (13). This is a serious objection to the procedure, as it may considerably increase the computing time for the larger values of k or m.

5. Use of orthogonal polynomials. In Sections 3 and 4 we have seen two difficulties inherent in the generation of $y_k(x)$ in the form (1). In this section we shall see that the use of orthogonal polynomials will solve both of the difficulties. Part of the advantage is expounded, for example, by Milne [7]. The author's attention was called to these matters several years ago by Professor Foreman Acton.

Suppose we have a set of polynomials $p_i(x) (i = 0, \cdots, k)$ with the property that

(14) $p_i(x)$ is a polynomial in x of proper degree i,

i.e., $p_i(x)$ is of degree i, but not of degree $i - 1$. Relation (14) implies that any polynomial of degree k in x is uniquely representable as a linear combination $c_0p_0(x) + \cdots + c_kp_k(x)$. Hence the $y_k(x)$ of (1) which is to fit the data values f_μ best may be represented in the form

$$y_k(x) = s_0^{(k)}p_0(x) + \cdots + s_k^{(k)}p_k(x). \tag{15}$$

The development in Sections 2 or 9 can be paralleled to determine $s_0^{(k)}, \cdots, s_k^{(k)}$ so that

$$\Psi(s_0^{(k)}, \cdots, s_k^{(k)}) = \sum_{\mu=1}^{m} \left\{ f_\mu - \sum_{h=0}^{k} s_h^{(k)} p_h(x_\mu) \right\}^2 = \text{minimum}. \tag{16}$$

The result will now be stated.

Introduce abbreviations analogous to those of g_{ij} and γ_i :

$$w_{ij} = \sum_{\mu=1}^{m} p_i(x_\mu)p_j(x_\mu); \qquad \omega_i = \sum_{\mu=1}^{m} f_\mu p_i(x_\mu). \tag{17}$$

Just as we derived (11), we derive the following normal equations which determine the correct $s_i^{(k)}$ uniquely:

$$\begin{array}{c} w_{00}s_0^{(k)} + \cdots + w_{0k}s_k^{(k)} = \omega_0 \\ \cdots\cdots\cdots\cdots\cdots \\ w_{k0}s_0^{(k)} + \cdots + w_{kk}s_k^{(k)} = \omega_k . \end{array} \tag{18}$$

Now the system (18) is very general, since the $p_i(x)$ satisfy only the conditions (14). In order that (18) be easily solvable for larger values of k, it is sufficient to make the off-diagonal elements $w_{ij}(i \neq j)$ considerably smaller than the diagonal elements w_{ii}. This is frequently done in practice by selecting for the $p_i(x)$ polynomials which are *orthogonal* with respect to some mass distribution. In a data-fitting code [8] written at the Lockheed Aircraft Company, for example, $p_i(x)$ was selected to be the i^{th} Chebyshov polynomial $T_i(x)$ over an interval containing all the x_μ. If the interval is $(-1, 1)$, one has

$$\int_{-1}^{1} T_i(x)T_j(x)(1 - x^2)^{-\frac{1}{2}}\,dx = 0,$$

so that one might expect the w_{ij} to be relatively small.

It is a usual practice to put the origin at the approximate midpoint of the x_μ, even though one uses the powers $p_i(x) = x^i$. Since for $i + j$ odd one then has

$$\int_{-a}^{a} x^i x^j\,dx = 0,$$

it is to be expected that w_{ij} will be relatively small for $i + j$ odd. This has the effect of approximately "decoupling" the system (18) into two subsets of linear equations.

The purpose of this section is to call the reader's attention to the advantages of having polynomials $p_i(x)$ for which

$$w_{ij} = \sum_{\mu=1}^{m} p_i(x_\mu)p_j(x_\mu) = 0. \tag{19}$$

Such polynomials are said to be *orthogonal over the point set* $x_1, \cdots, x_m$.

When (16) and (19) hold, the system (18) assumes the simple form

$$\begin{aligned} w_{00}s_0^{(k)} \qquad\qquad\qquad\qquad &= \omega_0 \\ w_{11}s_1^{(k)} \qquad\qquad &= \omega_1 \\ \cdots\cdots \qquad\qquad & \\ w_{kk}s_k^{(k)} &= \omega_k . \end{aligned} \tag{20}$$

(Note that the system (20) is truly decoupled.) Hence $s_h^{(k)} = s_h = \omega_h/w_{hh}$ depends only on h, and not on k. This is the important consequence of the use of polynomials $p_i(x)$ which are actually orthogonal over the set $x_1, \cdots, x_m$.

Let us re-examine Sections 3 and 4 in the light of (20). For the orthogonal polynomials $p_i(x)$, there is no longer any difficulty in solving the normal equations: the solution $s_h = \omega_h/w_{hh}$ is obtained with one division. Also, the fact that $s_h = s_h^{(k)}$ is independent of k removes the main difficulty mentioned in Section 4—that one would have to recompute σ_{k+1}^2 without taking advantage of σ_k^2. For note the following:

$$\begin{aligned} \delta_k^2 &= \sum_{\mu=1}^{m}\left\{f_\mu - y_k(x_\mu)\right\}^2 = \sum_{\mu=1}^{m}\left\{f_\mu - \sum_{h=0}^{k} s_h p_h(x_\mu)\right\}\left\{f_\mu - \sum_{i=0}^{k} s_i p_i(x_\mu)\right\} \\ &= \sum_{\mu=1}^{m} f_\mu^2 - 2\sum_{h=0}^{k} s_h \sum_{\mu=1}^{m} f_\mu p_h(x_\mu) + \sum_{h,i=0}^{k} s_h s_i \sum_{\mu=1}^{m} p_h(x_\mu)p_i(x_\mu) \\ &= \sum_{\mu=1}^{m} f_\mu^2 - 2\sum_{h=0}^{k} s_h\omega_h + \sum_{h=0}^{k} s_h^2 w_{hh} \qquad \text{(by (19) and (17))}, \\ &= \sum_{\mu=1}^{m} f_\mu^2 - \sum_{h=0}^{k} w_{hh}s_h^2; \end{aligned}$$

i.e., we have here the following analog of Parseval's identity for the ordinary theory of Fourier series:

$$\delta_k^2 = \sum_{\mu=1}^{m} f_\mu^2 - \sum_{h=0}^{k} w_{hh}s_h^2. \tag{21}$$

It follows from (21) that

$$\delta_k^2 = \delta_{k-1}^2 - w_{kk}s_k^2,$$

so that σ_{k+1}^2 may be computed directly from σ_k^2. In fact,

$$\sigma_{k+1}^2 = \frac{\delta_{k+1}^2}{m-k-2} = \frac{\delta_k^2 - w_{k+1,k+1}s_{k+1}^2}{m-k-2},$$

or

$$\sigma_{k+1}^2 = \frac{m-k-1}{m-k-2}\sigma_k^2 - \frac{w_{k+1,k+1}s_{k+1}^2}{m-k-2}. \tag{22}$$

It may be that formula (21) is a poor way to estimate δ_k^2 accurately, because of the cancellation of nearly equal terms. Nevertheless, (22) should provide a reasonable way to estimate σ_{k+1}^2 from σ_k^2. In any case, one has the $s_0, \cdots, s_k$, and can use these to determine $\sigma_0^2, \cdots, \sigma_k^2$.

6. Generation of orthogonal polynomials. At this point we have to say where we obtain the orthogonal polynomials used in Section 5. They may be obtained in many ways—for example by a Gram-Schmidt orthogonalization of the powers $1, x, x^2, \cdots$, (see [9]). The fact that in principle the solution of the system (11) by a variant of Gaussian elimination yields orthogonal polynomials $p_i(x)$ has been developed by Rushton [10]. The use of the three-term recurrence suggested by Householder [4] and by Stiefel [5] is very promising. In this we generate the orthogonal polynomials as follows:

$$p_0(x) = 1; \tag{23_0}$$

$$p_1(x) = xp_0(x) - \alpha_1 p_0(x); \tag{23_1}$$

$$p_2(x) = xp_1(x) - \alpha_2 p_1(x) - \beta_1 p_0(x); \tag{23_2}$$

$$\cdots\cdots\cdots\cdots$$

$$p_{i+1}(x) = xp_i(x) - \alpha_{i+1}p_i(x) - \beta_i p_{i-1}(x) \qquad (i = 1, 2, \cdots). \tag{23_{i+1}}$$

Here the α_i and β_i are numbers chosen to make the orthogonality relations (19) hold. We shall prove by induction that this is possible.

If we let $p_{-1}(x) = 0$, relation (23_1) is the special case $i = 0$ of (23_{i+1}). Suppose, for an induction hypothesis, that $\alpha_1, \cdots, \alpha_i, \beta_1, \cdots, \beta_{i-1}$ have been chosen so that $p_0(x), \cdots, p_i(x)$ are pairwise orthogonal in the sense of equation (19). Let us see how to choose α_{i+1} and β_i in (23_{i+1}) so that

$$\sum_{\mu=1}^{m} p_{i+1}(x_\mu)p_j(x_\mu) = 0 \qquad (\text{for all } j = 0, 1, \cdots, i). \tag{24}$$

Set $x = x_\mu$ in (23_{i+1}). Multiply by $p_j(x_\mu)$ and add over $\mu = 1, \cdots, m$. One has

$$\begin{aligned}\sum_{\mu=1}^{m} p_{i+1}(x_\mu)p_j(x_\mu) = \sum_{\mu=1}^{m} x_\mu p_i(x_\mu)p_j(x_\mu) - \alpha_{i+1}\sum_{\mu=1}^{m} p_i(x_\mu)p_j(x_\mu) \\ - \beta_i \sum_{\mu=1}^{m} p_{i-1}(x_\mu)p_j(x_\mu).\end{aligned} \tag{25}$$

Now, for $j < i - 1$, we know by our induction hypothesis that the last two terms of (25) are 0. Moreover, since $xp_j(x)$ is a polynomial in x of degree $j + 1 < i$, we know that it can be expressed as a linear combination of polynomials $p_0(x), \cdots, p_{i-1}(x)$. But then the sum

$$\sum_{\mu=1}^{m} x_\mu p_i(x_\mu) p_j(x_\mu)$$

must be 0, since $p_i(x)$ is orthogonal to each of the polynomials $p_0(x), \cdots,$ $p_{i-1}(x)$, by the induction hypothesis. Hence we have proved that $p_{i+1}(x)$ defined by (23_{i+1}) is orthogonal to $p_0(x), \cdots, p_{i-1}(x)$, for any choices of α_{i+1} and β_i.

Now if we put $j = i$ in (25), we can see that, if

$$\alpha_{i+1} = \sum_{\mu=1}^{m} x_\mu \{p_i(x_\mu)\}^2 \Big/ \sum_{\mu=1}^{m} \{p_i(x_\mu)\}^2, \tag{26}$$

then $p_{i+1}(x)$ is orthogonal to $p_i(x)$. Moreover, if we put $j = i - 1$ in (25), we can see that, if

$$\beta_i = \sum_{\mu=1}^{m} x_\mu p_i(x_\mu) p_{i-1}(x_\mu) \Big/ \sum_{\mu=1}^{m} \{p_{i-1}(x_\mu)\}^2, \tag{27}$$

then $p_{i+1}(x)$ is orthogonal to $p_{i-1}(x)$.

Thus, if α_{i+1}, β_i are chosen according to (26) and (27), we can be sure that $p_{i+1}(x)$ is orthogonal to $p_0(x), \cdots, p_i(x)$. Thus the induction is complete, and can be carried on for all i up to where $p_i(x_\mu) = 0$ $(\mu = 1, \cdots, m)$. The break-down can be shown to occur first for $i = m - 1$, when (26) fails.

Taking (23_{i+1}) for $x = x_\mu$, multiplying through by $p_{i+1}(x_\mu)$, and adding for $\mu = 1, \cdots, m$, yields the identity

$$\sum_{\mu=1}^{m} \{p_{i+1}(x_\mu)\}^2 = \sum_{\mu=1}^{m} x_\mu p_i(x_\mu) p_{i+1}(x_\mu).$$

Hence we may compute β_i by an alternative formula,

$$\beta_i = \sum_{\mu=1}^{m} \{p_i(x_\mu)\}^2 \Big/ \sum_{\mu=1}^{m} \{p_{i-1}(x_\mu)\}^2. \tag{28}$$

Making use of abbreviations (17), we get the following formulas for α_{i+1}, β_i:

$$\alpha_{i+1} = \sum_{\mu=1}^{m} x_\mu \{p_i(x_\mu)\}^2 / w_{ii}; \tag{29}$$

$$\beta_i = w_{ii}/w_{i-1,i-1}. \tag{30}$$

Using (29), (30) in formulas (23), one can generate the orthogonal polynomials $p_i(x)$ recursively.

The same technique can be used to generate polynomials orthogonal with respect to a generalization of (24):

$$\sum_{\mu=1}^{m} w_\mu^2 p_{i+1}(x_\mu) p_j(x_\mu) = 0,$$

where the w_μ^2 are arbitrary positive weights. This would correspond to using a *weighted norm* $\| e \|$ which would alter (7) to read

$$\Phi(t_0^{(k)}, \cdots, t_k^{(k)}) = \sum_{\mu=1}^{m} w_\mu^2 \{f_\mu - y_k(x_\mu)\}^2. \tag{7'}$$

Dr. M. Weisfeld has called the author's attention to the fact that this construction can be generalized also to polynomial functions of several real variables.

7. Programming the computation for a digital computer. With the above information, the preparation of an automatic program for solving the least-squares data-fitting problem (7) should be straightforward, except for the questions of significant digits and round off.

A single quantity like α_1 is called a *scalar*, and will be stored in a storage cell with a name like S_1, S_2, $\cdots$, S_1', S_2', $\cdots$, or S_1'', S_2'', $\cdots$. We shall call a set of values like $x = [x_1, \cdots, x_m]$ or $f = [f_1, \cdots, f_m]$ a *vector*. A vector will require a much larger storage space, which will be given a label like V_1, V_2, $\cdots$.

For any two vectors $y = [y_1, \cdots, y_m]$, $z = [z_1, \cdots, z_m]$, we denote by (y, z) the *scalar product*

$$(y, z) = \sum_{\mu=1}^{m} y_\mu z_\mu .$$

and by yz the *componentwise product vector*

$$yz = [y_1 z_1, \cdots, y_m z_m].$$

Initially the vectors x, f and scalar k are read into the machine and stored as follows:

$$x = [x_1, \cdots, x_m] \text{ in } V_1,$$

$$f = [f_1, \cdots, f_m] \text{ in } V_2,$$

$$k \text{ in } S_0 .$$

Let the vectors

$$p^{(-1)} = [p_1^{(-1)}, \cdots, p_m^{(-1)}] = [0, \cdots, 0] \quad \text{in} \quad V_3,$$

$$p^{(0)} = [p_1^{(0)}, \cdots, p_m^{(0)}] = [1, \cdots, 1] \quad \text{in} \quad V_4$$

be filed in temporary storage as the values at $x_1, \cdots, x_m$ of the polynomials $p_{-1}(x)$ and $p_0(x)$. Let the scalars

$$w_{00} = m \quad \text{in} \quad S_1,$$

$$\beta_0 = 0 \quad \text{in } S_0'$$

be in temporary storage.

The routine starts at step (31):

(31) Compute $\delta_{-1}^2 = (f, f)$ and store it in S_3.
(32) Put $i = 0$.
(33) Compute $\omega_i = (f, p^{(i)})$ and store it in S_4.
(34) Compute $s_i = \omega_i / w_{ii}$ and store it in S_{8+i}.
(35) Compute $\delta_i^2 = \delta_{i-1}^2 - s_i^2 w_{ii}$ and store it in S_5.
(36) Compute $\sigma_i^2 = \delta_i^2/(m - i - 1)$ and store it in S_{1+i}''.
(37) Test the accuracy of the approximation by comparing σ_i^2 with $\sigma_0^2, \cdots, \sigma_{i-1}^2$. (See Section 4.)
(38) If the approximation is close enough, or if $i \geqq k$, exit. If not, go on to step (39).
(39) Compute the vector $xp^{(i)} = [x_1 p_1^{(i)}, \cdots, x_m p_m^{(i)}]$ and store it in V_5.
(40) Compute $\alpha_{i+1} = (xp^{(i)}, p^{(i)})/w_{ii}$ and store it in S_{2i+1}'.
(41) Compute the vector $p^{(i+1)} = (x - \alpha_{i+1})p^{(i)} - \beta_i p^{(i-1)}$ and store it in V_5.
(42) Compute $w_{i+1,i+1} = (p^{(i+1)}, p^{(i+1)})$ and store it in S_7.
(43) Increase from i to $i + 1$. (Thus, among other steps, move $p^{(i)}$ to V_3, $p^{(i+1)}$ to V_4, and $w_{i+1,i+1}$ to S_1.)
(44) Compute $\beta_i = w_{ii}/w_{i-1,i-1}$ and store it in S_{2i}'.
(45) Return to step (33).

Thus we have saved the values $s_0, s_1, \cdots, \alpha_1, \alpha_2, \cdots, \beta_0, \beta_1, \beta_2, \cdots,$ and $\sigma_0^2, \sigma_1^2, \cdots$, while the other quantities are erased in the course of the computation. Now the routine determines in step (37) or (38) the order k of the polynomial $y_k(x)$ which is to fit the data. One will ordinarily then want to generate the value of this polynomial for various values of x, probably including $x_1, \cdots, x_m$. In principle one could use the s_i, α_i, and β_i to compute the coefficients c_i of the powers of x in $y_k(x)$, and then evaluate $y_k(x)$ from these coefficients. It seems likely that these c_i would grow rapidly as k grows, and that therefore it would be necessary to compute the c_i with very great precision. To avoid this difficulty it is recommended that one compute the $p_i(x)$ for each desired value of x from the recurrence (23), and simultaneously compute $y_k(x)$ from (15). It is believed that such a calculation will prove much less troublesome.

Such a routine would closely parallel the previous one, and would proceed as follows: Suppose the desired x's are $x_1, \cdots, x_m$. The vector x and scalar k are read into the machine and stored as follows:

$$\begin{aligned} x &= [x_1, \cdots, x_m] && \text{in } V_1, \\ y^{(-1)} &= [0, \cdots, 0] && \text{in } V_2, \\ & k \text{ in } S_0. \end{aligned}$$

Let the vectors

$$\begin{aligned} p^{(-1)} &= [0, \cdots, 0] && \text{in } V_3, \\ p^{(0)} &= [1, \cdots, 1] && \text{in } V_4 \end{aligned}$$

be filed. Assume that $s_0, \cdots, s_k, \alpha_1, \cdots, \alpha_k, \beta_0, \cdots, \beta_{k-1}$ are stored. Then:

(46) Put $i = 0$.
(47) Compute $s_i p^{(i)}$ and store it in V_6.
(48) Compute $y^{(i)} = y^{(i-1)} + s_i p^{(i)}$ and store it in V_2.
(49) If $i \geqq k$, exit. Otherwise, go on to step (50).
(50) Compute the vector $xp^{(i)}$ and store it in V_5.
(51) Compute the vector $p^{(i+1)} = (x - \alpha_{i+1})p^{(i)} - \beta_i p^{(i-1)}$, and store it in V_5.
(52) Increase from i to $i + 1$.
(53) Return to step (47).

The formulas (23) insure that the leading term of $p_i(x)$ is x^i, no matter what values $x_1, \cdots, x_m$ may have. Unless the x_μ have special properties one will find that the values $p_i(x_\mu)$ will become very large or small. While this can sometimes be taken of by appropriate scaling (see Section 8), in most machine codes the significance can most easily be preserved when the various $p_i(x_\mu)$ remain in the same range. For x_μ reasonably uniformly distributed throughout an interval $[a, b]$, the magnitude of the $p_i(x_\mu)$ should be close to that of the Legendre polynomials over the interval $[a, b]$. Since the Chebyshov polynomials $T_i(x)$ are a little simpler and have the same approximate magnitude as the Legendre polynomials, we will consider them.

On the interval $[-1, 1]$ for $i \geqq 1$, $T_i(x) = \cos[i(\text{arc cos } x)] = 2^{i-1}x^i + \cdots$, and $\max_{-1 \leqq x \leqq 1} | T_i(x) | = 1$. Hence $2^{1-i}T_i(x) = x^i + \cdots$ is normalized like the polynomials $p_i(x)$ of (23), and

$$\max_{-1 \leqq x \leqq 1} | 2^{1-i}T_i(x) | = 2^{1-i} \qquad (i \geqq 1).$$

Such an exponential decrease in the size of the $p_i(x)$ might have serious consequences for the routines described above. We may adapt the polynomials $T_i(x)$ to the interval $[-a, a]$ by writing

$$T_i(x/a) = 2^{i-1}(x/a)^i + \cdots .$$

When $a = 2$ we see that all polynomials $T_i(x/2)$ have leading coefficient $\frac{1}{2}$.

If we scale and shift our data points $x_1, \cdots, x_m$ so they approximately fill the interval $[-2, 2]$, *we may expect that the polynomial values $p_\mu^{(i)}$ of* (39) *will remain of approximately uniform size.* Hence the basic interval $-2 \leqq x \leqq 2$ is recommended for fixed-point codes using the routines of this paragraph.

8. SWAC codes. In the Mathematics Department of the University of California, Los Angeles, Mrs. Marcia Ascher is coding the above procedures for the digital computer SWAC. The following two arbitrary limitations have been placed on the parameters:

$$m \leqq 1023; \qquad k \leqq 32.$$

To take care of scaling, any vector $[z_1, \cdots, z_m]$ is stored in $m + 1$ cells in the *floating vector* form $[\zeta_0, \zeta_1, \cdots, \zeta_m]$. Here ζ_0 is an integer with $-2^{36} + 1 \leqq \zeta_0 \leqq 2^{36} - 1$, while

$$|\zeta_i| < 1 \qquad\qquad (\text{for all } i = 1, \cdots, m),$$

and $\frac{1}{2} \leqq \max_{1 \leqq i \leqq m} |\zeta_i| < 1$. One interprets the vector as follows:

$$z_i = \zeta_i \cdot 2^{\zeta_0}.$$

Each scalar is stored in 2 cells as a floating vector with $m = 1$ components.

The floating vector convention has been used as a successful scaling arrangement by Professor M. R. Hestenes in several matrix codes on SWAC [11].

It is planned to report on the success of Mrs. Ascher's codes in a later paper.

9. Proof that (7) has a unique solution. Let us define these column vectors (the x_μ are assumed distinct):

$$f = \begin{pmatrix} f_1 \\ \vdots \\ f_m \end{pmatrix}; \qquad t = t^{(k)} = \begin{pmatrix} t_0 \\ \vdots \\ t_k \end{pmatrix}; \qquad q^{(i)}, = \begin{pmatrix} x_1^{\,i} \\ \vdots \\ x_m^{\,i} \end{pmatrix}$$

$$(i = 0, 1, \cdots, k).$$

Let $Q = [q^{(0)} q^{(1)} \cdots q^{(k)}]$ be a matrix of m rows and $k + 1$ columns. Define $\|\cdot\|_2$ as in (4). Let T denote matrix transposition. With this notation, we can express (7) as the search for a t such that

$$\|f - Qt\|_2^2 = \text{minimum}. \tag{54}$$

LEMMA. *Assume $m > k$. Then, for any column vector c of $k + 1$ components, $c^T Q^T Q c \geqq 0$. Moreover, $c^T Q^T Q c = 0$ if and only if $c = 0$.*

PROOF. The square matrix Q^TQ is of order $k + 1$. Now $c^TQ^TQc = (Qc)^TQc = \| Qc \|_2^2 \geqq 0$. Suppose $c^TQ^TQc = 0$. Then $Qc = 0$, and this means that the polynomial $c_0 + c_1x_\mu + c_2x_\mu^2 + \cdots + c_kx_\mu^k$ vanishes for the m abscissas $x_1, \cdots, x_m$. Since $m > k$, the polynomial must be identically zero, i.e., $c = 0$.

This proves the lemma.

THEOREM. *If $m > k$, the vector $t = (Q^TQ)^{-1}Q^Tf$ is the unique vector minimizing* (54).

PROOF.

$$\begin{aligned}\| f - Qt \|_2^2 &= (f - Qt)^T(f - Qt)\\ &= f^Tf - t^TQ^Tf - f^TQt + t^TQ^TQt\\ &= f^Tf - 2t^TQ^Tf + f^TQ^TQt\\ &= f^Tf - 2t^Tg + t^TGt,\end{aligned}$$

where we have introduced the abbreviations $g = Q^Tf$ and $G = Q^TQ$.

Now the homogeneous system $Gc = 0$ has only the solution $c = 0$. For, otherwise, one would have $c^TGc = c^TQ^TQc = 0$ for $c \neq 0$, and this would contradict the lemma. It then follows from the theory of linear systems that any system $Gt = r$ has a unique solution. That is, the inverse matrix G^{-1} exists.

Continuing,

$$(55) \qquad \| f - Qt \|_2^2 = (t - G^{-1}g)^TG(t - G^{-1}g) + f^Tf - g^TG^{-1}g.$$

Now, by the lemma, the term $(t - G^{-1}g)^TG(t - G^{-1}g)$ is minimized when and only when $t = G^{-1}g$. Since the other terms on the right-hand side of (55) are independent of t, this proves the theorem.

The minimizing vector $t = G^{-1}g$ is the solution of the system $Gt = g$, which is the vector representation of the normal equations (11). Thus the present section completely replaces Section 2.

Note that the same proofs could be carried through when the columns of Q form an arbitrary linearly independent set. In particular, the i-th column of Q could be the values $p_i(x_\mu)$ of the orthogonal polynomial system of Section 5. Thus the present section also supplies the proof of (18).

REFERENCES

[1] E. P. NOVODVORSKIĬ AND I. Š. PINSKER, *On a process of equalization of maxima*, Uspehi Matem. Nauk, vol. 6, no. 6 (1951), pp. 174–181. (Russian)

[2] S. S. WILKS, *Mathematical statistics*, Princeton Univ. Press, 1943, 284 pp. (esp. pp. 166 ff.).

[3] C. LANCZOS, *Spectroscopic eigenvalue analysis*, J. Washington Acad. Sci., vol. 45 (1955), pp. 315–323.

[4] A. S. Householder, *Principles of numerical analysis*, McGraw-Hill, New York-Toronto-London, 1953, 274 pp. (p. 221).

[5] E. L. Stiefel, *Kernel polynomials in linear algebra and their numerical applications*, multilithed report, National Bureau of Standards, Washington, 1955, 52 pp.

[6] I. R. Savage and E. Lukacs, *Tables of inverses of finite segments of the Hilbert matrix*, pp. 105–108 of O. Taussky (editor), *Contributions to the solution of systems of linear equations and the determination of eigenvalues*, National Bureau of Standards Applied Mathematics Series 39, U. S. Govt. Printing Office, 1954, 139 pp.

[7] W. E. Milne, *Numerical calculus*, Princeton Univ. Press, 1949, 393 pp.

[8] Lockheed Aircraft Co., *A curve fitting code for the IBM Type 701 computer.*

[9] P. Davis and P. Rabinowitz, *A multiple purpose orthonormalizing code and its uses*, J. Assoc. Comput. Machinery, vol. 1 (1954), pp. 183–191.

[10] S. Rushton, *On least squares fitting by orthonormal polynomials using the Choleski method*, J. Roy. Stat. Soc. (B), vol. 13 (1951), pp. 92–99.

[11] M. R. Hestenes, manuscript in preparation.

[12] D. Hilbert, *Ein Beitrag zur Theorie des Legendre'schen Polynoms*, Acta Math., vol. 18, (1894), pp. 155–160.

University of California, Los Angeles*
The Ramo-Wooldridge Corporation
Los Angeles

*On September 1, 1957, the author transferred to Stanford University.